Remarks

In the final Office Action of January 25, 2005, the Examiner rejects claims 21, 22, and 24-26 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,477,562 to Nemirovsky ("Nemirovsky"); rejects claims 1-4, 6-9, and 23 under 35 U.S.C. § 103(a) under 35 U.S.C. § 103(a) in view of Nemirovsky and an excerpt from Computer Organization and Design, The Hardware/Software Interface, by John Hennessy et al. ("Hennessy (Computer Organization and Design)"); and rejects claims 10-20 and 28 under 35 U.S.C. § 103(a) in view of Nemirovsky, and an excerpt from Computer Architecture A Quantitative Approach, Second Edition, by David Patterson and John Hennessy ("Hennessy (Computer Architecture)") and in view of Hennessy (Computer Organization and Design); and rejects claims 27, 29, and 30 under 35 U.S.C. § 103(a) in view of Nemirovsky and Hennessy (Computer Architecture).

By this Amendment, Applicants propose amending claims 1, 10, 21, 24 and 27 to more appropriately define the invention and canceling claim 31 without prejudice or disclaimer. Support for the amendment to claims 1, 10, and 27 can be found, for example, at paragraphs 42-46 of the originally filed specification. Support for the amendments to claim 21 can be found, for example, in paragraph 59 of the originally filed specification.

CLAIM REJECTIONS UNDER 35 U.S.C. § 102(e) BASED ON NEMIROVSKY

Claims 21, 22, and 24-26 stand rejected under 35 U.S.C. § 102(e) based on Nemirovsky. Applicants respectfully traverse this rejection.

A proper rejection under 35 U.S.C. § 102 requires that a single reference teach every aspect of the claimed invention either explicitly or impliedly. Any feature not directly taught must be inherently present. See M.P.E.P. § 2131. Nemirovsky does not disclose or suggest the combination of features recited in Applicants' claims 21, 22, and 24-26.

Claim 21, as amended, is directed to a method for processing a packet to determine control information for the packet. The method includes reading a plurality of instructions, generating a predicted address based on a predetermined one of the read instructions, evaluating the read instructions by applying operations specified in the read instructions to at least a portion of the packet, and selecting one of the read instructions based on the evaluations. Further, the method includes performing operations related to determining the control information for the packet based on the selected instruction, the operations including generating a next address for reading instructions.

Nemirovsky is directed to prioritized instruction scheduling for multi-streaming processors. The multi-streaming processor of Nemirovsky has multiple streams for processing multiple threads. (Nemirovsky, Abstract). According to Nemirovsky, the various streams are assigned priority codes, which are used to determine relative access of the streams to resources as well as which stream has access at any point in time. (Nemirovsky, Abstract).

Although Nemirovsky discloses the execution of multiple instruction streams, the multiple instruction streams appear to generally be independent of

one another and are executed on a priority basis. This aspect of Nemirovsky is described in detail at column 7, lines 12-27. Nemirovsky states:

> The net effect of the queues is that there are concurrent streams of instructions from which eligible instructions may be issued to functional resources. Each stream that the processor is equipped to execute has a context frame containing the program counter and register file for that stream. A thread is made active by loading an available context frame with the thread's program counter address and register values and by assigning it an active priority. There may be only a single thread to be executed, in which case there is a single stream of instructions to execute. When there are more active threads than streams available to execute threads, a number of threads up to the available number of context frames are made active and the remaining threads remain temporarily inactive. It is typically a function of an operating system to assign threads to streams of a multi-streaming processor.

(Nemirovsky, col. 7, lines 12-27). As described in this section, each instruction stream has a program counter and a register file. When a thread is to be made active, its "context" is loaded and the thread begins to execute. Once a thread is selected and made active, the instructions in the thread appear to execute in a conventional manner.

Claim 21, in contrast to Nemirovsky, discloses, for example, reading a plurality of instructions, evaluating the read instructions by applying operations specified in the read instructions to at least a portion of a packet, and selecting one of the read instructions based on the evaluations. Nemirovsky completely fails to disclose or suggest evaluating read instructions by applying operations specified in the read instructions to at least a portion of a packet and selecting one of the instructions based on the evaluations, as recited in claim 21.

In rejecting claim 21, the Examiner points to column 6, line 65 through column 7, line 11 of Nemirovsky as disclosing the evaluation of read instructions

and selecting one of the instructions based on the evaluations, stating that this section shows "each queue is partitioned into units for each stream and thus the individual instructions are inherently evaluated to see what stream they belong to." (Office Action, numbered paragraphs 6 and 42). This cited section of Nemirovsky relates to fetching instructions from memory, where the fetched instructions may be "stored in prefetch buffers, decoded and placed in one or more queues."

Applicants respectfully disagree with the Examiner's statement that "individual instructions are inherently evaluated to see what stream they belong to." Applicants submit that it is not inherent that an instruction needs to be evaluated to determine the instruction stream or queue to which it belongs. Nemirovsky, for example, states that "a queue may be dedicated to one or a set of resources." (Nemirovsky, col. 7, lines 9-10). Therefore, it is not inherent that an instruction needs to be evaluated to be assigned to a particular resource.

In any event, Applicants submit that claim 21, as proposed to be amended, is not disclosed by Nemirovsky. Claim 21, for example, recites evaluating the read instructions by applying operations specified in the read instructions to at least a portion of the packet. Nemirovsky does not disclose oor suggest this feature of claim 21. Nemirovsky, at column 7, lines 7-11, discloses at a high level how instructions may be assigned to queues. Nothing in this section of Nemirovsky appears to relate to evaluating instructions by applying operations specified in the read instructions to at least a portion of the packet. Nemirovsky does disclose functional units 207-210 that eventually evaluate

instructions. (Nemirovsky, Fig. 2 and column 7, lines 43-47). These functional units 207-210, however, appear to evaluate the instructions after they are assigned to queues/streams of Nemirovsky. Accordingly, functional units 207-210 can not be said to evaluate instructions that are then selected based on the evaluations.

For at least these reasons, Applicants submit that the rejection of claim 21 is improper and should be withdrawn. At least by virtue of its dependency on claim 21, the rejections of claim 22 and 24-26 under 35 U.S.C. § 102(e) are also improper and should be withdrawn.

Additionally, dependent claims 22 and 24-26 recite additional features that are not disclosed or suggested by Nemirovsky. Amended claim 24, for example, further defines claim 21 and recites that the selection of one of the read instructions is performed as a priority selection among ones of the read instructions that evaluate to a logic true value. The Examiner points to column 7, lines 57-60, and column 5, lines 10-21, of Nemirovsky as allegedly disclosing the features of claim 21. (Office Action, numbered paragraph 8). Although these sections of Nemirovsky generally discuss using multiple instruction streams and selecting among the streams for access to resources, nothing in these sections of Nemirovsky discloses or suggests the selecting an instruction as a priority selection based on a read instruction that evaluates to a logic true value. These sections of Nemirovsky do not disclose a priority selection among instructions, much less a priority selection among instructions that evaluate to a logic true value. The Examiner appears to interpret any instruction that is selected as

corresponding to a logic true value. (Office Action, numbered paragraph 8). However, the priority selection of claim 24, as amended, is recited as being performed from among ones of the read instruction that evaluate to a logic true value.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103(A) BASED ON NEMIROVSKY AND HENNESSY (COMPUTER ORGANIZATION AND DESIGN)

Claims 1-4, 6-9, and 23 stand rejected under 35 U.S.C. § 103(a) based on Nemirovsky in view of Hennessy (Computer Organization and Design). Applicants respectfully traverse this rejection. More particularly, in rejecting independent claim 1, the Examiner contends that Nemirovsky discloses many of the features recited in claim 1 but concedes that Nemirovsky does not disclose a plurality of memories nor pipelining. The Examiner contends, however, that one of ordinary skill in the art would have found it obvious to modify Nemirovsky in view of Hennessy to obtain the claimed invention. Applicants disagree and respectfully traverse this rejection.

Claim 1, as amended, is directed to a pipelined processor comprising a first pipeline stage and a second pipeline stage. The first pipeline stage includes a plurality of instruction memories and a program counter corresponding to a location in each of the instruction memories from which instructions are read. The second pipeline stage includes an evaluation component corresponding to each of the instruction memories, the evaluation component generating evaluation results based on each of the instructions read from the instruction

corresponding to a logic true value. (Office Action, numbered paragraph 8). However, the priority selection of claim 24, as amended, is recited as being performed from among ones of the read instruction that evaluate to a logic true value.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103(A) BASED ON NEMIROVSKY AND HENNESSY (COMPUTER ORGANIZATION AND DESIGN)

Claims 1-4, 6-9, and 23 stand rejected under 35 U.S.C. § 103(a) based on Nemirovsky in view of Hennessy (Computer Organization and Design). Applicants respectfully traverse this rejection. More particularly, in rejecting independent claim 1, the Examiner contends that Nemirovsky discloses many of the features recited in claim 1 but concedes that Nemirovsky does not disclose a plurality of memories nor pipelining. The Examiner contends, however, that one of ordinary skill in the art would have found it obvious to modify Nemirovsky in view of Hennessy to obtain the claimed invention. Applicants disagree and respectfully traverse this rejection.

Claim 1, as amended, is directed to a pipelined processor comprising a first pipeline stage and a second pipeline stage. The first pipeline stage includes a plurality of instruction memories and a program counter corresponding to a location in each of the instruction memories from which instructions are read. The second pipeline stage includes an evaluation component corresponding to each of the instruction memories, the evaluation component generating evaluation results based on each of the instructions read from the instruction

of the program counter of claim 1, which is recited as corresponding to a location in each of the instruction memories from which instructions are read. Instead, one or ordinary skill in the art modifying Nemirovsky to include multiple instruction memories would likely modify Nemirovsky to independently read from the instruction memories based on the program counters corresponding to each of the streams. Nothing in Nemirovsky or Hennessy discloses or suggests a program counter corresponding to a location in a plurality of instruction memories, as required by claim 1.

At numbered paragraph 13(c) of the Office Action, the Examiner appears to state that the disclosure of Hennessy (Computer Organization and Design) of a first level cache and a second level cache corresponds to the plurality of instruction memories recited in claim 1. Applicants respectfully disagree with the Examiner's interpretation of Hennessy (Computer Organization and Design). A first and second level cache is not equivalent to multiple instruction memories. As is generally known in the art, and as is disclosed by Hennessy (Computer Organization and Design) at page 576, a multi-level cache is used to enhance access to an underlying memory. At page 576, Hennessy (Computer Organization and Design) specifically discloses use of a multi-level cache to enhance the access speed for DRAMs. Thus, if anything, the multi-level cache of Hennessy (Computer Organization and Design) should be considered part of the underlying DRAM, and in no way discloses or suggests a plurality of instruction memories.

Arguments similar to the above arguments were presented in the previous Amendment. In numbered paragraph 44 of the Office Action, the Examiner addresses this argument, stating that "the fact that Nemirovsky has a single program counter for each stream would not disallow a 103 combination with a reference teaching a program counter corresponding to a location in each of a plurality of instruction memories unless each stream had its own independent instruction memory." Although the Examiner may be correct in the statement that Nemirovsky does not explicitly "disallow" various modifications, the fact that a reference does not positively disallow a modification does not make the modification obvious under 35 U.S.C. § 103. As stated above, there is no teaching or suggestion in Nemirovsky to use a program counter corresponding to a location in each of the instruction memories, as recited in claim 1. If anything, because Nemirovsky is explicitly directed to a multi-processing stream system having independent processing streams, Nemirovsky teaches away from using a single program counter for a plurality of the processing streams. Accordingly, Applicants submit that the Examiner has not presented a proper *prima facie* case of obviousness regarding this feature of claim 1.

Claim 1, as amended, further recites an evaluation component corresponding to each of the instruction memories and a priority encoder configured to select one of the instructions based on evaluation results generated from the instructions read from the instruction memories, where the instructions not selected by the priority encoder are discarded. Neither Nemirovsky nor Hennessy (Computer Organization and Design), either alone or in combination,

disclose or suggest this feature of claim 1. As noted by the Examiner, Nemirovsky discloses selecting instructions from different streams for execution. (see Office Action, numbered paragraph 42). In Nemirovsky, non-selected instructions appear to simply be delayed for selection at a later time. (see Nemirovsky, col. 7, lines 23-25). In stark contrast, as currently recited in claim 1, the instructions not selected by the priority encoder are discarded. Applicants submit that nothing in Nemirovsky or Hennessy (Computer Organization and Design) disclose or suggest this feature of the invention.

For at least these reasons, the rejection of claim 1 based on Nemirovsky or Hennessy (Computer Organization and Design) is improper and should be withdrawn. Claims 2-4 and 6-9 depend from claim 1. At least by virtue of their dependency from claim 1, Applicants submit that the rejection of these claims is also improper and should be withdrawn.

Dependent claim 23 was also rejected under 35 U.S.C. § 103(a) based on Nemirovsky and Hennessy (Computer Organization and Design). Applicants submit that Hennessy (Computer Organization and Design) does not cure the above-mentioned deficiencies of Nemirovsky as discussed for claim 21, from which claim 23 depends. Accordingly, the rejection of claim 23 is improper and should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103(a) BASED ON NEMIROVSKY IN VIEW OF HENNESSY(COMPUTER ARCHITECTURE)

Claims 27, 29, and 30 stand rejected under 35 U.S.C. § 103(a) based on Nemirovsky and Hennessy (Computer Architecture). Applicants respectfully traverse this rejection.

Claim 27, as amended, is directed to a processing device including means for simultaneously reading a plurality of processing instructions from instruction memory that relate to processing a packet, the device further includes means for selecting one of the read instructions for execution based on a priority encoding of evaluation results related to each of the read instructions, and discarding non-selected ones of the read instructions.

Applicants submit that Nemirovsky does not disclose or suggest the features recited in amended claim 27. For instance, Nemirovsky does not disclose or suggest means for selecting one of the read instructions for execution based on a priority encoding of evaluation results related to each of the read instructions, where the non-selected ones of the read instructions are discarded. Nemirovsky discusses a number of possible schemes for assigning priorities to instruction streams. None of these techniques, however, involve selecting one of a plurality of read instructions based on a priority encoding of evaluation results related to each of the read instructions. Nemirovsky selects a stream and then evaluates instructions within the selected stream. Nemirovsky does not disclose, however, selecting an instruction (or stream) based on evaluation results related to each of a plurality of instructions. Moreover, as previously discussed, Nemirovsky does not disclose or suggest discarding non-selected instructions. In stark contrast, Nemirovsky discloses that non-selected threads are

"temporarily inactive." (Nemirovsky, column 7, lines 23-25). Applicants respectfully submit that Hennessy (Computer Architecture) does not disclose or suggest this deficiency in Nemirovsky.

For at least these reasons, Applicants submit that the rejection of claim 27 is improper and should be withdrawn. At least by virtue of its dependency on claim 27, the rejections of claims 29 and 30 under 35 U.S.C. § 103(a) are also improper and should be withdrawn.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103(a) BASED ON NEMIROVSKY IN VIEW OF HENNESSY(COMPUTER ARCHITECTURE) AND HENNESSY (COMPUTER ORGANIZATION AND DESIGN)

Claims 10-20 and 28 stand rejected under 35 U.S.C. § 103(a) based on Nemirovsky, Hennessy (Computer Architecture), and Hennessy (Computer Organization and Design). Applicants respectfully traverse this rejection.

Claim 10, as amended, is directed to a network device that comprises a physical interface and a processing unit. The processing unit includes a pipelined packet processing engine that includes: a first pipeline stage configured to read a plurality of packet processing instructions relating to processing of a first packet from instruction memories per processing cycle, and a second pipeline stage configured to select one of the packet processing instructions for execution and discard the non-selected ones of the plurality of packet processing instructions.

In rejecting claim 10, the Examiner contends that Nemirovsky discloses many of the features recited in claim 10, but concedes that Nemirovsky does not

disclose the first or second pipeline stages recited in claim 10. (Office Action, numbered paragraph 24). For these features, the Examiner relies on Hennessy (Computer Organization and Design) to disclose a multi-stage pipeline and Hennessy (Computer Architecture) to disclose reading a plurality of instructions per cycle.

Although Hennessy (Computer Organization and Design) and Hennessy (Computer Architecture) may generally disclose a pipeline and the execution of multiple instructions simultaneously, neither Hennessy (Computer Organization and Design) nor Hennessy (Computer Architecture), discloses or suggests the second pipeline stage recited in claim 10, which is configured to select one of the packet processing instructions for execution and discard the non-selected ones of the plurality of packet processing instructions. The superscalar processor of Hennessy (Computer Architecture), for instance, is described as a system that allows multiple instructions to issue in a clock cycle and can execute multiple instructions in parallel. Hennessy (Computer Architecture), however, does not disclose or suggest selecting an instruction and discarding the non-selected instructions. Hennessy (Computer Organization and Design) also does not disclose or suggest this feature of claim 10. Further, as previously discussed in reference to claim 1, Nemirovsky also fails to disclose or suggest discarding non-selected instructions, as recited in claim 10.

For at least these reasons, Applicants submit that Nemirovsky, Hennessy (Computer Organization and Design), and Hennessy (Computer Architecture), either alone or in combination, fail to disclose or suggest each element of claim

10. Accordingly, the rejection of claim 10 based on these references is improper and should be withdrawn. Claims 11-20 depend from claim 10, either directly or indirectly. The rejection of these claims is also improper at least by virtue of their dependency on claim 10.

Dependent claims 11-20 recite additional features that are not disclosed or suggested by the applied references. Claim 12, for instance, recites a program counter configured to store a program address value used to read the packet processing instructions from the instruction memories. The Examiner points to Nemirovsky as allegedly disclosing this feature of the invention. (Office Action, numbered paragraph 26). As previously discussed, Applicants submit that although Nemirovsky discloses the use of a program counter, Nemirovsky does not disclose or suggest a program counter used to read instructions from the instruction memories. The instructions recited in claim 12 refer to the plurality of instructions recited in claim 10. Nemirovsky, in contrast, appears to use a separate program counter for each instruction stream, and thus does not disclose or suggest the program counter recited in claim 12. Accordingly, for this reason also, the rejection of claim 12 is improper and should be withdrawn.

Claim 17 recites that the evaluation components generate the evaluation results based on a logical operation dictated by the packet processing instructions. The Examiner points to column 6, lines 50-60 of Nemirovsky as allegedly disclosing this feature. The section of Nemirovsky generally discloses the operation of priority control unit 9. Although Nemirovsky discloses that priority control unit 9 uses "logic" to switch among scheduling mechanisms,

priority control unit 9 does not generate results based on a logical operation dictated by packet processing instructions. The priority logic of control unit 9 appears to instead be based on criteria such as "a history of processing activity." (Nemirovsky, column 6, lines 53-54). Accordingly, for this additional reason, the rejection of claim 17 is improper and should be withdrawn.

Dependent claim 28 was also rejected under 35 U.S.C. § 103(a) based on Nemirovsky, Hennessy (Computer Architecture), and Hennessy (Computer Organization and Design). Applicants submit that Hennessy (Computer Architecture) and Hennessy (Computer Organization and Design) do not cure the above-mentioned deficiencies of Nemirovsky as discussed for claim 27, from which claim 28 depends. Accordingly, the rejection of claim 28 is improper and should be withdrawn.

Applicant respectfully requests that this Amendment under 37 C.F.R. § 1.116 be entered by the Examiner, placing claims 1-4 and 6-30 in condition for allowance. Applicants respectfully point out that the final action by the Examiner presented some new arguments as to the application of the art against Applicants' invention. Additionally, Applicants submit that the entry of the amendment would place the application in better form for appeal, should the Examiner dispute the patentability of the pending claims.

In view of the foregoing amendments and remarks, Applicants respectfully request the Examiner's reconsideration of this application, and the timely allowance of the pending claims.

To the extent necessary, a petition for an extension of time under 37 CFR 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 50-1070 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY & SNYDER, L.L.P.

By: _____
Brian E. Ledell
Reg. No. 42,784

11240 Waples Mill Road
Suite 300
Fairfax, Virginia 22030
(571) 432-0800

Date: March 25, 2005